浪脚本零起点入门系列(九)

maxscript 函数

作者：飞浪

函数 function

函数作为程序中不可分割的一部分,扮演着极其重要的角色。maxscript 中的函数定义如下：

(function | fn) <name> { <parameter> } = <expr>

例如一个简单的函数：

```
fn p = print localtime
```

运行之后得到：p(),这样这个函数就定义成功了，调用的时候你只需要执行一下 p()，就会运行此函数"="号后面的内容，此函数的意思是输出当前时间。

接下具体来看，fn 是声明这是一个函数，就跟 rollout 表示创建一个栏一样。fn 是缩写，全写是 function 就是函数的意思，所以刚才那个函数也可以写成:

```
function p = print localtime
```

p 是函数的名字，"="号后面是函数内容，可以是任意代码块（block）。所以一个函数至少需要四个部分，即声明 fn，函数名称，"="号和后面的代码。

可以这样理解，所谓函数就是能完成一定功能的代码块，只不过把它写成函数后可以随便调用，像 print 也是一个函数，它的功能是 maxscript 已经写好了的。

再看下面这个函数：

```
fn ranMtlColor col =
(
if classof col == color then
for i in meditmaterials where classof i == standardmaterial do i.diffuse = col
)
```

这个函数多了一个部分，就是函数名称 ranMtlColor 加了个参数 col，这个 col 是函数的一个变量，只是个形式参数，它只在此函数内有效，运行此函数，得到 ranMtlColor ()表示创建函数成功，然后调用函数：

```
ranMtlColor red
```

这次调用的时候函数名称后面带了一个参数，这是个实际参数，red 是颜色值红色，当然你也可以用(color 255 0 0)来表示。

这个函数的意义是把材质编辑器中的所有 standard 材质的固有色调成红色。

这样，你就可以随意修改颜色了，如

```
ranMtlColor blue
```

ranMtlColor (color 34 64 123)

形式参数也可以先赋个值，即初始化。如：

```
fn ranMtlColor col:(color 255 0 0) =
(
if classof col == color then
for i in meditmaterials where classof i == standardmaterial do i.diffuse = col
)
```

创建这个函数后，因为 col 已经给了个初始值(color 255 0 0)，这样调用函数的时候可以直接执行 ranMtlColor()而不带参数，

这样就默认颜色(color 255 0 0)，如果想修改此参数则调用如下：

ranMtlColor col:(color 0 255 0)

return 表达式

有时候函数不一定要执行完才返回结果，这个我们可以用 retuen 提前返回，如：

```
fn myfunction num =
( if classof num == integer and num >= 1 then
 for i in 1 to num do
 ( if i == 50 then return "提前返回"
 )
 format "完全执行\n"
)
```

运行函数，然后调用：

myfunction 10 --得到结果"完全执行"

myfunction 100 --得到结果"提前返回"，因为当 i 等于 50 的时候,直接跳出了函数返回"提前返回",所以 i 就在 50 的时候停止循环了，这样在计算大量数据时可以节约很多运算量。

当然函数也可以返回指定的结果，如：

```
fn calcN n =
( sum=0
 if classof n == integer and n >= 1 then
 (   for i = 1 to n do sum+=i
 )
 sum
)
```

这个函数是把正整数累加，即 1+2+3+...+n

如：执行 calcN 100，得到正确答案 5050，执行 calcN -1 得到 0，那么就可以弹出对话框说输入数据不规范了。

函数自身调用--递归函数

函数可以自己调用自己，看下面的例子：

```
fn ranMtlColor mat col =
(
 if classof mat == Standardmaterial then mat.diffuse = col
 else if classof mat == Multimaterial then
 for j in mat do ranMtlColor j col --调用自己
 else ( )
)
```

这样做可以减少代码重复，并且循环到最里面的元素。这个函数作用是把多维子材质的所有 standard 子材质固有色改成指定颜色。

或者你觉得可以一句搞定没必要这么复杂，但是你看一下下面这个材质：

```
m = Multimaterial ()
m[1]=Multimaterial ()
meditMaterials[1] = m
```

多维子材质套多维子材质，还可以再往里加，如果用这个函数：

```
ranMtlColor meditMaterials[1] red
```

一下搞定里面所有的 standard 材质。

递归函数可以用在材质转换，文件夹读取等等脚本里面。

函数可以互相调用，但要注意的是，在脚本里面后面的函数可以调用前面的函数，前面的不能调用后面的，函数里面的变量都属于局部变量，当然特殊声明 global 除外。

函数道理不难懂，就是把你平时写的一段代码给安上一个名称，下次直接喊那个名字，就执行你那段代码，你可以把你在 rollout 里面的按钮都写成函数（当然是需要的时候），在 on ... pressed do 时直接调用函数。活用函数，往往能得到意想不到的效果。

本节就到这里，谢谢观看！

本节原帖地址：http://www.cgplusplus.com/bbs/viewthread.php?tid=520

本节结束。